

Niakwa Programming Language

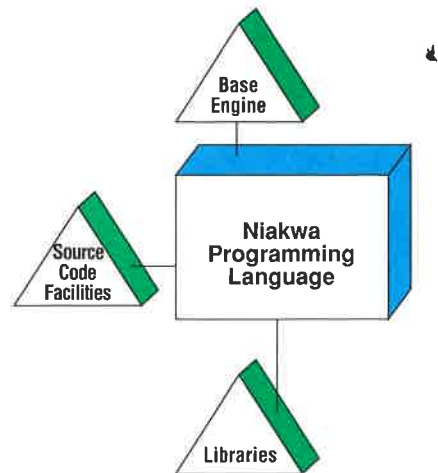
Introduction

Designed for the professional developer, the Niakwa Programming Language (NPL) offers a set of powerful features which allow for the development of innovative, highly competitive applications.

NPL accomplishes this by providing exceptional:

- Portability
- Ease of Learning and Maintenance
- Programmer Productivity

By incorporating powerful fourth-generation capabilities into a third-generation language, Niakwa provides software developers 100% true portability and flexibility.



NPL provides true source and object code compatibility across most major operating environments. This allows programs developed in one set of code to be run on a variety of hardware and operating system solutions.

In other words, professional software developers can greatly expand their target markets without spending time developing and supporting different versions of source code for each application.

In addition, developers (especially C programmers) find the structured format of NPL very easy to learn and maintain.

NPL allows programmers to incorporate modules ("black box" routines) into new or existing programs to increase functionality and performance. Development in NPL is often much faster than in other languages (i.e., C).

The use of Long Identifier Names (LINs) and program constructs also provides programmers greater flexibility to logically name and identify variables. This allows for easy program maintenance in a minimum amount of time.

By providing modular technology, LINs, program constructs, and many other convenient features, NPL significantly improves programmer productivity and, thus, company profitability.

This data sheet provides an orientation to the language with an NPL organizational overview and description of the features.

NIAKWA

The Structure of NPL

Software Structure

NPL consists of three physical software products which create and/or execute NPL pseudo-code (p-code):

NPL Interpreter

- Creates and executes p-code -
The interpreter (also known as the incremental compiler) efficiently develops and maintains NPL programs. Using "Immediate Mode," the interpreter supports real-time: source code editing, syntax checking, variable inspection, variable modification, program halting, single instruction stepping and trapping, instruction logic tracing, etc.

In addition, the interpreter's relatively small code size and high performance allow its development capabilities to be used for end user support without disturbing application performance.

NPL RunTime Program

- Executes p-code generated by the interpreter or compiler -
The RunTime program executes NPL applications at end user sites. Executing p-code only, the RunTime program does not allow for program inspection, modification, variable inspection, etc., as provided in the interpreter.

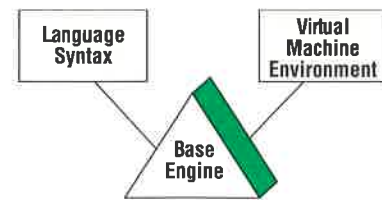
NPL Compiler

- Creates p-code -
The compiler primarily assists the interpreter with NPL program manipulation (conveniently performed on a batch basis).

Mechanical Structure

NPL can be further broken down into three main components:

- Base Engine
- Source Code Facilities
- Library Support



Base Engine

Language Syntax

NPL provides a robust 3-GL language suitable for development of sophisticated business or engineering applications.

Structured Language

NPL provides a set of powerful language features which allow for the development and use of modular technology.

FUNCTIONS/PROCEDURES

The FUNCTION/PROCEDURE interface provides a traditional, structured methodology for subroutine development and access.

Modules

Program modules allow for the development of true "black box" routines (application-independent libraries).

Program Constructs

NPL offers a collection of instrumental program constructs including:

- WHILE/WEND
- REPEAT/UNTIL
- IF/ELSE/END IF
- SWITCH/CASE
- FOR/BEGIN/NEXT

RECORDS/FIELDS

Using RECORDS/FIELDS, you can define logical record layouts. This, in conjunction with modules, permits a very high level of abstraction within database applications.

Programming Features

NPL's programming features provide for efficient data manipulation.

Advanced Math Functions

NPL provides an internal BCD numeric format that provides for fast arithmetic operations while providing accuracy to 13 digits.

NPL also provides advanced intrinsic math functions such as trig functions, logs, and exponentials. Use of a math coprocessor for these functions is supported on many environments.

NPL provides MAT math functions such as MAT multiply, MAT transposition, MAT identity, and MAT inverse.

Data Conversion

NPL supplies a series of data conversion statements, including: CONVERT, BIN, VAL, \$PACK, and \$UNPACK. These statements efficiently maximize code power and flexibility to convert data from one form to another.

String Handling

NPL provides sophisticated string manipulation functions such as MAT SEARCH and built-in sort and merge operations. NPL also provides low level operations that allow complete bit level control of strings. Strings can be dynamically expanded up to 2 GB on 32 bit platforms.

Multi-user

NPL supports up to 512 users and contains numerous critical features (i.e., unique terminal, task, and user identification, and file locking).

HELP Subsystem

Developers can integrate help functions into their applications to improve end user support.

Error Control

NPL offers error handling at the program, line, function, and command levels; and easy design of generic error traps and "user-proof" applications.

Demonstration Capabilities

NPL offers convenient, cost efficient capabilities for the generation and use of demonstration software.

Demo Diskette Feature

Niakwa provides discounted, limited-execution count RunTimes which allow for on-site application demonstration.

Self-running Demonstrations

\$DEMO generates "self-demonstrating" software and automatic end user training tutorials.

Non-English Development

NPL developers have a reputation for being very successful in the international marketplace as a result of NPL support for extensive character sets and localized application development.

Character Sets

NPL supports all standard, non-English character sets.

Localized Application Development

Because NPL supports non-English character sets, applications can easily be written in a variety of foreign languages and, thus, increase developer strength in international markets.

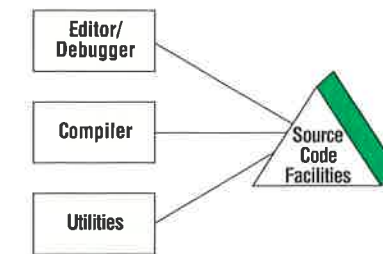
Documentation

Niakwa provides comprehensive easy-to-read documentation.

Virtual Machine Environment

NPL developers enjoy the flexibility of operating their software on a variety of operating systems and hardware solutions while still maintaining a consistent look and feel.

They are able to accomplish this task because of the many operating system-like specifications (i.e., workstation, disk device, printer, and communication controls) incorporated into NPL's language definition. No other language provides this Virtual Machine Environment.



Source Code Facilities

The NPL source code facilities are comprised of a powerful, integrated editor/debugger, incremental compiler, and comprehensive set of utilities.

Integrated Editor/debugger

NPL is one of the easiest languages to debug due to a series of powerful debugging commands; including, LIST, TRACE, PRINT, and STEP commands, etc. The combined use of NPL's interpreter and line editor make program editing fast and easy.

Incremental Compiler

As source program lines are entered, they are immediately compiled by the interpreter into executable p-code and stored in program text memory. The original source line is then effectively discarded. Whenever the original source line must be subsequently viewed (for listing, editing, debugging, etc.), it is regenerated from the stored p-code by a built-in de-compiler.

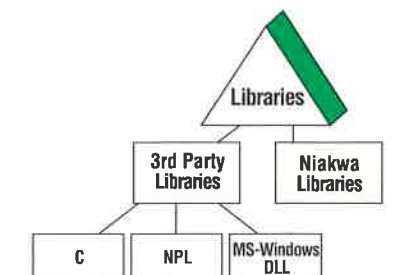
Utilities

NPL provides a comprehensive set of utilities for convenient program development, including:

- File Backup
- File Copy
- File Scratching
- File Recovery
- Diskimage Creation
- Diskimage Listing
- Diskimage Sizing
- Device Settings
- Edit Options, etc.

Program Encryption

NPL's "scramble-protected" mode protects your sensitive code.



Libraries

NPL's modular technology enables developers to create and use true application-independent library programs. Using these libraries, developers can easily integrate added functionality without devoting development time to writing routines.

NPL Developers enjoy access to both Niakwa and third-party libraries.

Niakwa Libraries

Available to all licensed NPL users, these libraries are developed by Niakwa and written in NPL.

Third-party Libraries

NPL developers can access a variety of third-party developed C, NPL, and MS-Windows Dynamic Linked Libraries (DLLs).

A selection of external subroutines written in C, PASCAL and ASSEMBLER can also be integrated into NPL applications.

This enables NPL Developers to increase execution speed for selected processor-intensive functions and access resources and features of a specific environment.

Niakwa publishes library availability in the Niakwa Library Catalog (distributed periodically) and on the Niakwa Bulletin Board System (BBS) Library Forum.

Developer Benefits

Use of NPL's efficient development features reduces development time and increases programmer productivity.

- **Interactive Environment**

NPL's interactive environment increases programmer productivity by eliminating the need for recompilation when editing programs.

- **Easy to Learn and Maintain**

NPL's incremental compiler technology makes it an easy language to learn, code, and debug. Programmers familiar with C find NPL particularly easy to learn because of the many C-like features incorporated into the language.

- **Fast Prototyping**

NPL allows for quick development and testing of application prototypes.

End user Benefits

NPL provides a constant interface between the virtual machine and native development environments which adds to the portability and longevity of NPL applications.

- **Superior Performance**

NPL's high-level functions quickly perform entire operations that, in other languages, would require significantly more code and time.

- **Source Code Compatibility**

NPL source code is fully compatible across all NPL supported platforms. This gives applications a permanent look and feel (regardless of the operating environment in use).

- **Absolute Portability**

Because of NPL's unparalleled portability, NPL applications can operate on practically any platform the user chooses.

- **Device Independence**

Sophisticated control of all native device types (i.e., terminals, printers, storage devices, serial I/O devices, etc.) is provided across all NPL-supported operating environments.

Supported Environments

AIX	Novell NetWare
Intel UNIX	OS/400
MS-DOS	SuperDOS
MS-Windows	VMS
NetBIOS	Xenix

Technical Support

Niakwa provides one of the most prompt and comprehensive support systems in the industry today. In 1993, 99% of all Niakwa support calls were responded to within one hour, with an average resolution time of 15 minutes or less.

Maximum Limits

Program Size	Limited only by physical memory
Lines/Module	32,118
Variables/Module	65,534
Data File Size	16 MB normal 4 GB extended